1. UVOD

U ovom specijalističkom radu baviću se projektovanjem, razvojem i implementacijom baze podataka vezane za konkretan studijski primjer edukacije pravosudnih institucija, konkretno sudovi i tužilaštva u Republici Srpskoj.

U samom radu obradiću pored samog modela baze i koncepte ciklusa odnosno faze razvoja projekta, te samu implementaciju korisničkog interfejsa preko objektno orijentisanog programskog jezika C#.

Ovaj projekat možemo definisati kao sređen skup metoda, procesa i operacija za prikupljanje, čuvanje, obradu, prenošenje i distribuciju podataka u okviru jedne organizacije, uključujući i korisnike koji se tim aktivnostima bave. Ovakav sistem je, u stvari, model realnog sistema. On opisuje i mijenjanje realnog sistema. Cilj je prikupljanje i obezbjeđivanje relevantnih informacija iz realnog sistema i okruženja i transformacija istih u upravljačke informacije radi održavanje sistema, odnosno upravljanje samim sistemom.

2. ŽIVOTNI CIKLUS PROJEKTA

Neka od pitanja koja se najčešće postavljaju kod razvoja novog softvera su: Kako se trebaju alocirati odgovornosti na klase i objekte, kako bi objekti trebali sarađivati, šta bi klase trebale da rade, itd. Ovo su kritična pitanja koja se postavljaju kod projektovanja sistema, a do odgovora na ta pitanja se treba doći primjenom Objektno orijentisane analize i dizajna (OOA/D). S druge strane, na osnovu starih rješenja se mogu iskazati najbolje prakse ili paterni, te se kao takvi mogu koristiti kod budućeg razvoja softvera.

2.1 Larman metodologija

Upravo Larmanova metodologija treba da prikaže kako se na efikasan način mogu primjeniti znanja iz OOA/D i paterna.

U Larmanovoj metodologiji možemo razlikovati 4 faze:

Prva faza je **faza prikupljanja zahtjeva** u kojoj se definišu svojstva i uslovi koje sistem ili šire gledajući projekat mora da zadovolji. Glavni izazov kod analize zahtjeva je da se pronađe, poveže i zapamti (najčešće da se zapiše) šta je zaista potrebno u obliku koji je jasan klijentima i članovima razvojnog tima.

Svi zahtjevi se prema FURPS+ modelu mogu podijeliti u funkcionalne (**F**unctional), zahtjeve vezane za upotrebljivosti (**U**sability), pouzdanost (**R**eliability), performanse (**P**erformance) i podrživost (**S**upportability). Veoma česta podjela je i ona na funkcionalne i nefunkcionalne zahtjeve.Funkcionalni zahtjevi definišu zahtjevane funkcije sistema, dok nefunkcionalni definišu sve ostale.

U ovom radu ćemo se fokusirati na funkcionalne zahtjeve i na njihovo definisanje.

Funkcionalni zahtjevi se uglavnom predstavljaju preko slučajeva korišćenja. Slučajevi korišćenja predstavljaju tekstualnu priču koja se koristi za pronalaženje i pamćenje zahtjeva.

Za prikazivanje svih slučajeva korišćenja se koristi Model slučajeva korišćenja.On prestavlja skup svih slučajeva korišćenja, aktora i veza između aktora i slučajeva korišćenja.Aktor u tom kontekstu predstavlja nešto što ima ponašanje,bez obzira da li je to osoba, kompjuter ili organizacija. Model slučajeva korišćenja može opciono da uključuje i UML case dijagrame za prikazivanje imena slučajeva korišćenja i aktora i njihovih